

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of : Greg HETHERINGTON
Filed : with an effective filing date of April 22, 1998
For : METHOD AND APPARATUS FOR
PROCESSING FREE-FORMAT DATA
Group Art Unit : 2172
Examiner : Ella Colbert
Docket : GRIHAC P22AUSD1

The Commissioner of Patents and Trademarks
Washington, D.C. 20231

FIRST PRELIMINARY AMENDMENT

Dear Sir:

By way of preliminary amendment, please amend the above identified application as set forth below.

In the Specification:

Please cancel the first full paragraph on page 28 of the specification, in its entirety, in favor of a clean form of the first full paragraph on page 28 of the specification, without any markings thereon, as follows. Also accompanying this Preliminary Amendment is a copy of the original paragraphs of the specification which show the addition(s) (by underlining) and the deletion(s) (by brackets) to the canceled specification paragraph. Please enter the replacement specification paragraph into the record of this case.

In the Claims:

Please cancel claims 1 - 53, without prejudice or disclaimer of the subject matter therein, in favor of new claims 54 - 106 as follows.

In the Abstract:

Please cancel the Abstract of the Disclosure, presently on file, and enter into the record of this application the new Abstract of the Disclosure typed on a separate sheet and accompanying this Preliminary Amendment.

The text object (illustrated in Figure 3) comprises a plurality of component nodes 302-312. The next object can be represented as a text node tree, having branches (e.g. 313) wherein the component nodes 302-312 are positioned in a predetermined hierarchy. The "lowest" hierarchy is at the bottom of the text node tree and the "highest" hierarchy is at the top of the text node tree. The node 302 at the top of the node tree will be referred to as the "root node. It will be appreciated that components of the text object can be stored in any convenient manner in a memory of a processing means, could be nested within each other, for example, refer to each other in some way, etc. The text object is able to be represented as a text node tree, but that does not mean that it is stored in memory in this way. As long as the components of the text object can be processed in such a fashion that the components act like component nodes of a text node tree as represented in the figure, then this is sufficient.

Claims:

54. (NEW) A method of processing free-format data stored in a computing system, comprising the steps of examining elements of the data to determine attributes of the data, by examining the content of the elements and the contextual relationships of elements to each other, to determine semantic and syntactic information (attributes) about the data, producing additional data relating to this information, in the form of a text object which includes pointer means enabling access to the elements of the free-format data, and the additional data being accessible by a query processing means to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

55. (NEW) The method according to claim 54, wherein the free-format data is stored as a record in a free-format field of a database.

56. (NEW) The method according to claim 54, wherein the data remains stored in the computing system as it was originally stored, whereby it may be accessed by other applications.

57. (NEW) The method according to claim 54, wherein the text object includes an attribute - type identifier which identifies an attribute type of an element of the data.

58. (NEW) The method according to claim 54, wherein the text object includes a value indicating the character length of an element of the data.

59. (NEW) The method according to claim 57, wherein the text object includes a value indicating whether an element is a low level in a syntactic hierarchy or higher level whereby the value may be used for matching purposes when matching data with other data processed in accordance with the method.

60. (NEW) The method according to claim 54, wherein the text object including a match weighting value for an element of the data, which can be used to determine the significance of the element when matching with other free format data.

61. (NEW) The method according to claim 54, wherein the text object comprises a plurality of component nodes arranged according to the semantic structure of the free-format data, the component nodes being arranged in a hierarchy corresponding to the semantic structure of the free-format data and each component node including additional data relating to the corresponding element of the free-format data.

62. (NEW) The method according to claim 54, comprising the further step of generating matching values for comparing an element of the free-format data with an element of other free-format data processed in accordance with the present method.

63. (NEW) The method according to claim 62, wherein the matching value is a phonetic value for phonetically comparing elements of free-format data.

64. (NEW) The method according to claim 54, wherein the text object includes implied data relating to information implied from the free-format data.

65. (NEW) The method according to claim 54, wherein a plurality of free-format data records are processed and a text object associated with each free-format data record is produced.

66. (NEW) The method according to claim 65, wherein a plurality of free-format data records are processed and a text object associated with each free-format data record is produced.

67. (NEW) The method according to claim 65, comprising the further step of producing a text object index including attribute type identifiers for elements of each data record and pointers to each data record, whereby the index may be queried by queries relating to semantic and syntactic information about the data and the data may be accessed via the index.

68. (NEW) The method according to claim 67, wherein each entry in the text object index includes a representative value key, which gives a value representative of the element associated with the attribute - type identifier.

69. (NEW) The method according to claim 54, comprising the further step of carrying out a domain construction process to construct a domain object being arranged to carry out the examination process by parsing the free-format data in accordance with grammar rules.

70. (NEW) The method according to claim 69, wherein the domain definition data files include character definition data, regular expression definition data and grammar data.

71. (NEW) The method according to claim 54, wherein the free-format data is postal address data.

72. (NEW) The method according to claim 54, the query processing means can carry out normal database operations on the data via the additional data.

73. (NEW) A processing system for processing free-format data stored in a computing system, the apparatus including means for examining elements of the data to determine attributes of the data, by examining the content of the elements and the contextual relationships of elements to each other, to determine semantic and syntactic information (attributes) about the data, means for producing additional data relating to this information, in the form of a text object which includes pointer means enabling access to the elements of the free-format data, and a query processing means which is arranged to access the additional data to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

74. (NEW) The processing system according to claim 73, wherein the free-format data is stored as a record in a free-format field of a database.

75. (NEW) The processing system according to claim 73, wherein the examining means does not affect the storage of the data.

76. (NEW) The processing system according to claim 73, wherein the text object includes an attribute - type identifier which identifies an attribute type of an element of the data.

77. (NEW) The processing system according to claim 73, wherein the text object includes a value indicating the character length of an element of the data.

78. (NEW) The processing system according to claim 76, wherein the text object includes a value, indicating whether an attribute - type of an element is low level in a syntactic hierarch or high level whereby the value may be used for matching purposes when matching with other free-format data processed in accordance with this system.

79. (NEW) The processing system according to claim 73, wherein the text object includes a match weighting value for an element of the data, which can be used to determine the significance of the element when matching with other free-format data.

80. (NEW) The processing system according to claim 73, wherein the text object comprises a plurality of component nodes arranged according to the semantic structure of the free-format data, the component nodes being arranged in a hierarchy corresponding to the semantic structure of the free-format data, and each component node including additional data relating to the corresponding element of free-format data.

81. (NEW) The processing system according to claim 73, wherein the text object means for generating matching values for comparing an element of the free-format data with an element of other free-format data processed by the processing system.

82. (NEW) The processing system according to claim 81, wherein the matching value is a phonetic value for phonetically comparing elements of free-format data.

83. (NEW) The processing system according to claim 73, wherein the text object includes implied data relating to information implied from the free-format data.

84. (NEW) The processing system according to claim 73, wherein the system is arranged to process a plurality of free-format data records and produce a text object associated with each free-format data record.

85. (NEW) The processing system according to claim 84, wherein the means for producing additional data is arranged to produce a text object index including attribute - type identifiers for elements of each data record and pointers to each data record and wherein the query processing means is arranged to access the text object index to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

86. (NEW) The processing system according to claim 85, wherein the text object index includes representative value keys for entries, which give a value representative of a feature of the element associated with the attribute - type identifier for the entry for facilitating matching with other free-format data processed in accordance with this system.

87. (NEW) The processing system according to claim 73, further comprising a domain object, the domain object being arranged to carry out the examination process by parsing the free-format data in accordance with grammar rules.

88. (NEW) The processing system according to claim 87, wherein the domain object is produced by a domain construction process from domain definition data files.

89. (NEW) The processing system according to claim 88, further comprising a domain constructor for carrying out the domain construction process.

90. (NEW) The processing system according to claim 88, wherein the domain definition data files include character definition data, regular expression definition data and grammar data.

91. (NEW) The processing system according to claim 73, wherein the free-format data is postal address data.

92. (NEW) The processing system according to claim 73, wherein the query processing means is arranged to carry out normal database operations on the data via the additional data.

93. (NEW) A method of enabling access to free-format data stored in a computing system, including a plurality of free-format data records, comprising the steps of storing additional data relating to semantic and syntactic information (attributes) about the data for each data record, the additional data being in the form of a text object associated with each data record, the text object including pointer means enabling access to elements of each free-format data record, the additional data being accessible by a query processing means to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

94. (NEW) A processing system for enabling access to free-format data stored in a computing system, including a plurality of free-format data records, the processing system comprising additional data relating to semantic and syntactic information (attributes) about the data for each data record, stored and accessible by the processing system, the additional data being in the form of a text object associated with each data record, the text object including pointer means enabling access to elements of each free-format data record, and a query processing means arranged to access the additional data to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

95. (NEW) A method of enabling access to free-format data stored in a computing system, including a plurality of free-format data records, comprising the steps of storing additional data relating to semantic and syntactic information (attributes) about the data of each data record, the additional data being in the form of a text object index which includes attribute - type identifiers for elements of each data

record and pointers to each data record, the text object index being accessible by a query processing means to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

96. (NEW) A processing system for enabling access to free-format data stored in a computing system, including a plurality of free-format data records, the processing system comprising the additional data relating to semantic and syntactic information (attributes) about the free-format data for each data record, the additional data being in the form of a text object index which includes attribute type identifiers for elements of each data record and pointers to each data record, and a query processing means arranged to access the additional data to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

97. (NEW) The method of accessing free-format data processed according to claim 54, comprising the steps of accessing the additional data to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

98. (NEW) The processing system for enabling access to free-format data processed according to the method of claim 54, the processing system including a query processing means arranged to access the additional data and provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data to manipulate the data.

99. (NEW) A processing system for processing free-format data stored in a computing system, comprising means for examining elements of the data to determine attributes of the data, by examining the content of the elements and the contextual relationship of elements to each other, to determine semantic and syntactic information (attributes) about the data, and a query processing means for utilizing this information

to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data.

100. (NEW) The processing systems according to claim 99, wherein the examining means retains the free-format data as stored in the computer system, without affecting it.

101. (NEW) A method of processing free-format data stored in a computing system, comprising the steps of examining elements of the data to determine attributes of the data, by examining the content of the elements and the contextual relationships of elements to each other, to determine semantic and syntactic information (attributes) about the data, and querying the data using this information to provide answers to queries relating to the semantic and syntactic information about the data and/or to access the data.

102. (NEW) The method of processing free-format data in accordance with claim 101, wherein the free-format data is unaffected by the examining process and remains stored in the computing system as it was originally stored.

103. (NEW) The computer readable memory storing instructions for controlling a computer to process free-format data stored in a computing system, according to the method of claim 54.

104. (NEW) The computer readable memory storing instructions for controlling a computer to process free-format data stored in a computing system, according to the method of claim 101.

105. (NEW) A method of processing a plurality of records of free-format data stored in a computing system, comprising the steps of, for each record, examining elements of the data to determine attributes of the data, by examining the content of the elements and the contextual relationships of elements to each other, to determine semantic and syntactic information (attributes) about each record, and producing virtual

data fields associated with each record enabling access to this information and the associated elements, whereby each record is provided with associated virtual data fields enabling access to semantic and syntactic information about that record and also access to the associated elements.

106. (NEW) A processing system for processing free-format data records stored in a computing system, comprising means for examining elements of the data of each record to determine attributes of the data, by examining the content of the elements and the contextual relationship of elements to each other, to determine semantic and syntactic information (attributes) about the data, and means for producing virtual data fields associated with each record enabling access to this information and the associated elements, whereby each record is provided with associated virtual data fields enabling access to semantic and syntactic information about that record and also access to the associated elements.

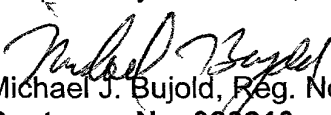
REMARKS

Accompanying this Preliminary Amendment, please find replacement paragraphs and marked-up paragraphs of the specification which overcome some informalities noted in the specification on file. The undersigned avers that the enclosed replacement paragraph(s) of the specification do not contain any new matter.

Please enter the above before consideration of this application. With respect to the above newly entered claims, the subject matter of the Chapter II amended claims is editorially revised and rewritten to bring that subject matter into conformity with the United States claim format.

In the event that there are any fee deficiencies or additional fees are payable, please charge the same or credit any overpayment to our Deposit Account (Account No. 04-0213).

Respectfully submitted,


Michael J. Bujold, Reg. No. 32,018
Customer No. 020210
Davis & Bujold, P.L.L.C.
Fourth Floor
500 North Commercial Street
Manchester NH 03101-1151
Telephone 603-624-9220
Facsimile 603-624-9229
E-mail: patent@davisandbujold.com

Marked-up Version of Paragraph

The text object (illustrated in Figure [1] 3) comprises a plurality of component nodes 302-312. The next object can be represented as a text node tree, having branches (e.g. 313) wherein the component nodes 302-312 are positioned in a predetermined hierarchy. The "lowest" hierarchy is at the bottom of the text node tree and the "highest hierarchy is at the top of the text node tree. The node 302 at the top of the node tree will be [refer] referred to as the "root node. It will be appreciated that components of the text object can be stored in any convenient manner in a memory of a processing means, could be nested within each other, for example, refer to each other in some way, etc. The text object is able to be represented as a text node tree, but that does not mean that it is stored in memory in this way. As long as the components of the text object can be processed in such a fashion that the components act like component nodes of a text node tree as represented in the figure, then this is sufficient.

FIG. 3 is a diagram illustrating a text node tree structure. The diagram shows a root node at the top, which branches into several child nodes. These child nodes further branch into more child nodes, creating a hierarchical tree structure. The nodes are represented by small circles, and the branches are represented by lines connecting the nodes. The tree structure is shown in a vertical orientation, with the root node at the top and the leaf nodes at the bottom.

Abstract of the Disclosure

A method and apparatus for processing free-format data (301) to produce a “text object” associated with the free-format data. The text object comprises a plurality of “component nodes” (302-312) containing attribute-type identifiers for elements of the free-format text and other data facilitating access to the text object to obtain information and change or add the free-format data. This arrangement obviates the need for the provision of separate database fields for each element of the information. Free-format data can therefore be processed in a similar manner to the way a human being processes free-format data. All elements can be accessed via the constructed text object.